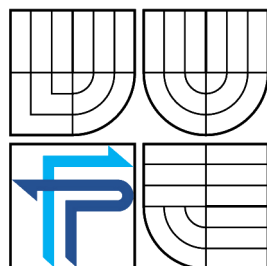


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FAKULTY OF BUSINESS AND MANAGEMENT
DEPARTMENT OF INFORMATICS

ANALÝZA A OPTIMALIZACE DATABÁZOVÉHO SYSTÉMU DATOVÉHO CENTRA SOCIÁLNÍCH SLUŽEB LIBERECKÉHO KRAJE

ANALYSIS AND OPTIMIZATION OF DATABASE SYSTEM OF DATA CENTER OF
SOCIAL SERVICES IN LIBEREC REGION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ LAUERMAN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ KŘÍŽ , Ph.D.

BRNO 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Lauerman Ondřej

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

**Analýza a optimalizace databázového systému Datového centra sociálních služeb
Libereckého kraje**

v anglickém jazyce:

**Analysis and Optimization of Database System of Data Center of Social Services in Liberec
Region**

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Seznam odborné literatury:

CONOLLY, T., BEGG, C., HOLOWCZAK, R. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. Brno: Computer Press, 2009. 584 s. ISBN 978-802-5123-287.

KOCH, M.; NEUWIRTH, B. Datové a funkční modelování. 3. vyd. Brno: Akademické nakladatelství CERM, 2008. 121 s. ISBN 978-802-1437-319.

LAUERMAN, J.; BENEŠOVÁ, I. Krajská metodická příručka plánování sociálních služeb v Libereckém kraji. Praha: Vysoká škola regionálního rozvoje, 2010. 80 s. ISBN 978-80-87174-01-2.

LAUERMAN, J. Vstupní analýza podmínek pro vytvoření Datového centra sociálních služeb Libereckého kraje. Liberec: JL-plán, 2008. 25 s.

Zákon č.108/2006 Sb., o sociálních službách, ze dne 14. března 2006.

Vedoucí bakalářské práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Děkan fakulty

V Brně, dne 17.05.2012

ABSTRAKT

Cílem této práce je analýza podpůrného databázového systému, který je využíván Datovým centrem Libereckého kraje a návrhy na jeho optimalizaci s ohledem na praktické použití zejména finančních ukazatelů.

ABSTRACT

The purpose of this bachelor's thesis is to analyze supporting database system used by Data Center of Social Services in Liberec Region, and suggest steps to optimize it for taking advantage of financial pointers.

KLÍČOVÁ SLOVA

databáze, SQL, entity, normalizace

KEYWORDS

database, SQL, entities, normalization

LAUERMAN, Ondřej. *Analýza a optimalizace databázového systému Datového centra sociálních služeb Libereckého kraje*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2012. 43 s. Vedoucí bakalářské práce Ing. Jiří Kríž, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.
Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 30. května 2012

.....

OBSAH

Úvod	9
1 Vymezení problému a cíle práce	10
1.1 Vymezení problému	10
1.2 Cíl práce	12
2 Teoretická východiska práce	13
2.1 Historie databázových systémů	13
2.2 Současné databázové systémy	14
2.3 Relační model	14
2.3.1 Entitně-relační modelování	16
2.3.2 Normalizace	17
2.3.3 Další normální formy	18
3 Analýza problému a současné situace	19
3.1 Požadavky na vytvoření Datového centra	19
3.2 Realizace a podoba Datového centra	21
3.3 Analýza použitého řešení	21
3.3.1 Značení tabulek	21
3.3.2 Struktura číselníků	22
3.3.3 Struktura datových tabulek	24
4 Vlastní návrhy řešení	26
4.1 Normalizace	26
4.1.1 Dodržení první normální formy	26
4.1.2 Dodržení druhé normální formy	29
4.1.3 Dodržení třetí normální formy	29
4.2 Označení tabulek	30
4.3 Přejmenování atributů	30
4.4 Řízení přístupů k DB	31

Závěr	38
Seznam použité literatury	39
Seznam symbolů, veličin a zkratk	43

ÚVOD

Don't just make random changes. There really are only two acceptable models of development: "think and analyze" or "years and years of testing on thousands of machines". Those two really do work. [12]

Linus Torvalds

V dnešní době jsou databázové systémy nezanedbatelnou součástí snad všech představitelných odvětví, které mají alespoň minimální spojitost s informačními technologiemi. Usnadňují nám shromažďování a evidenci nejrůznějších dat a jejich následné zpracování. Jsou na ně proto díky širokému využití kladeny vysoké nároky, zejména na způsob uložení dat a jejich následné snadné zpracování. Staly se také častým předmětem podnikání vývoje softwarových společností.

Návrh správného a funkčního databázového systému musí splňovat určitá kvalitativní kritéria. Zda tomu tak ve výsledku opravdu je ukáže zpravidla čas ověřený vhodnou analýzou, nebo „roky a roky testování strojů“ [12]. Proto jsem se při zpracování bakalářské práce snažil o co nejlépe provedenou a co nejvíce použitelnou analýzu databázového systému, která by byla použitelná pro zvýšení použitelnosti Datového centra.

1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

1.1 Vymezení problému

V naší republice existuje mnoho sociálních služeb provozovaných jak státem, tak soukromými subjekty. Mnoho z nich se ve svém poli působnosti překrývá, a potřebným se naopak nedostává potřebné péče. Podle Zákona č. 108 o sociálních službách se rozumí „sociální službou činnost nebo soubor činností podle tohoto zákona zajišťujících pomoc a podporu osobám za účelem sociálního začlenění nebo prevence sociálního vyloučení“ [7, §3 bod a)]. Tento zákon vymezuje tři základní druhy služeb, a to sociální poradenství, služby sociální péče a služby sociální prevence. Mezi tyto sociální služby a zařízení, která je poskytují, patří například [10]:

- domovy pro seniory
- dětské domovy
- protidrogová centra
- pečovatelské služby
- azylové domy
- noclehárny
- terénní programy
- ...

V systému sociálních služeb působí celá řada aktérů, kteří mají různé potřeby a požadavky v oblasti práce s daty a informacemi. Mezi nejdůležitější aktéry patří:

- poskytovatelé služeb (tj. organizace, které zajišťují přímý výkon a tyto služby poskytují),
- zřizovatelé (zadavatelé) služeb (např. stát, kraj, obec, církev, apod., tj. instituce, které zadávají práci poskytovatelům a zpravidla ji také financují – zcela nebo z části),

- uživatelé služeb (klienti),
- samosprávné orgány obcí a krajů,
- úřady plnící své zákonem dané povinnosti v sociální oblasti (obecní úřady, úřad práce, apod.).

Celý systém tedy zahrnuje velké množství dat a informací různého charakteru (evidenční, kapacitní, ekonomické, metodické, atd.). V gesci Ministerstva práce a sociálních věcí (MPSV) vznikla v uplynulém období celá řada nástrojů, které jsou využívány jako datová podpora pro oblast sociálních služeb. Jedním z nedostatků je ale chybějící celková koordinace těchto aktivit, zejména z pohledu připravovaných změn kompetencí (přechod větší odpovědnosti a financování sociálních služeb ze státu na kraje).

Pro oblast sociálních služeb je zřízen celostátní registr, který má za úkol udržovat informace o všech těchto službách a zařízeních, a jejich vlastnostech (počet lůžek, využití, apod.). Tento registr sice shromažďuje mnoho údajů, které ale často nejsou aktuální a naopak chybí řada dalších údajů, které jsou potřebné pro vyhodnocování a účelné plánování služeb na regionální úrovni.

V Libereckém kraji došlo proto v rámci procesu střednědobého plánování rozvoje sociálních služeb k rozhodnutí vytvořit vlastní systém pro datovou podporu sociálních služeb. Ten by měl využít existující data, zajistit jejich průběžnou aktualizaci v provázanosti s centrálním registrem a doplnit koncentrovanými a přesnějšími údaji z celého Libereckého kraje. Rozšířená data měla přinést lepší kontrolu nad provozovanými zařízeními a následné posouzení rozdělování státních dotací a financí z evropských fondů, a především pro plánování do dalších let, rozhodování o výstavbě a provozu těchto zařízení, atd.

Byly proto vytvořeny požadavky na „Datové centrum služeb v LK“, které by umožňovalo splnit výše zmíněné zadání. Dokument „Vstupní analýza“ [5] obsahuje základní ideové a obsahové principy, na kterých bylo Datové centrum postaveno.

1.2 Cíl práce

Cílem této práce by mělo být zhodnocení návrhu databázového systému, na kterém je postaveno Datové centrum. Hlavními kritérii hodnocení bude splnění normálních forem, logická struktura dat, vhodné označení relačních tabulek a jejich atributů, použitelnost jednotlivých výstupních ukazatelů s ohledem na využití všemi zainteresovanými stranami v problematice sociálních služeb nejen v Libereckém kraji, ale v celé České republice.

V poslední části práce by mělo dojít k navržení konkrétních kroků, které by pomohly vylepšit aktuální stav, a vyřešit problémy nalezené v analytické části práce.

2 TEORETICKÁ VÝCHODISKA PRÁCE

2.1 Historie databázových systémů

Prvotní náznak databázových systémů se začal objevovat jako firemní snahy utřídit ručně vytvořené kartotékové systémy nesoucí informace o výrobcích, dodavatelích, odběratelích, a dalších obchodních partnerech, a všeobecně nesoucí libovolné seříděné informace. Tyto kartotéky byly pečlivě označeny, umístěny ve skříních, a zpravidla zabezpečeny pomocí zámku. Poskytovaly tak jednoduchý a efektivní způsob řízení přístupu k těmto informacím. Vyhledávání informací v těchto informacích probíhá lidskou silou, a jeho rychlost je přímo závislá na schopnostech jednotlivých pracovníků, kteří mají tyto kartotéky na starost. Úroveň těchto schopností ovšem klesá úměrně k rostoucímu množství uložených údajů. Tento trend lze částečně eliminovat použitím jednoduché indexace, ale už nelze použít robustnější metody pro vyhledávání dat v takto rozsáhlých systémech, jako jsou například křížové odkazy [1, s. 44].

Zároveň s rozvojem informačních technologií a s rostoucí potřebou uchovávat velké množství organizovaných dat vznikla v roce 1960 pracovní skupina Conference on Data Systems Languages (Codasyl), a to za přispění ministerstva obrany Spojených států amerických. Tato skupina měla jako hlavní cíl vytvořit programovací jazyk pro podporu obchodních aplikací, jehož výsledkem se stal jazyk Common Business-Oriented Language (COBOL). Ve skupině Codasyl pak došlo i k vytvoření podskupiny Database Task Group (DBTG), která ustanovila základní specifikaci programovacích jazyků (zejména COBOLu) pro práci s databázemi. Většina databázových systémů té doby, založených na principech definovaných skupinou Codasyl, používala síťový datový model, a o několik let později společnost IBM implementovala hierarchický datový model, a v roce 1970 se objevil první návrh, jak implementovat relační datový model za využití algebry a relačního kalkulu, jehož práce by byla možná s jednoduchým jazykem s výrazy založenými na angličtině [13].

Tento navržený relační model ale trpěl některými nedostatky, zejména omezenými modelovacími schopnostmi. Proto byl v roce uveden nový návrh tzv. entitně-relačního (ER) modelu, který se stal širokou veřejností přijatým standardem.

2.2 Současné databázové systémy

V dnešní době se pod pojmem databáze skrývá více významů. Původní označení databáze je pro úložiště dat, které kromě samotných dat uchovává i další informace o těchto datech, zejména metadata typu relační vztahy, různá integritní omezení, apod. Přístup k databázi pak zajišťuje část databáze známá jako systém řízení báze dat (SŘBD). Ten umožňuje uživateli přistupovat k datům uloženým v databázi bez nutné znalosti fyzické struktury dat a poskytuje mu tak nadstavbu nad těmito daty. Dále SŘBD hlídá přístup k databázi a autorizuje jednotlivé uživatele podle jim svěřených pravomocí.

Databáze lze rozdělit na několik základních modelů, které se od druhé poloviny 20. století postupně vyvíjely. Následuje jejich stručný přehled seřazený chronologicky [9]:

- model správy souborů
- hierarchický
- síťový
- relační
- post-relační
- objektový

Protože analyzovaný databázový systém je postaven na relačním datovém modelu, budu se na následujících stránkách věnovat pouze jemu.

2.3 Relační model

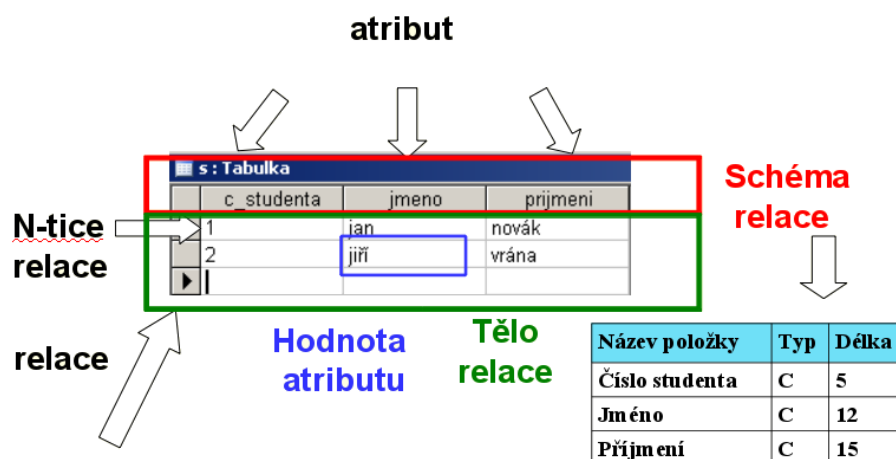
Relační datový model byl navržen E. F. Coddem, zaměstnancem společnosti IBM [13], a zveřejněn v roce 1970 v práci „A Relations Model of Data for Large Shared Data Banks“. Tento model si kladl 3 základní cíle [1, s. 62]:

- zajistit nezávislost dat na aplikačních programech, ty nesmějí být ovlivněny vnitřní strukturou uložených dat,
- umožnit použití tzv. normalizovaných relací,

- zpřístupnit jazyk pro manipulaci s daty.

Relační model se skládá z pěti hlavních složek [1, s. 63]:

- relace – „tabulka“, je tvořena schématem relace a tělem relace, jak ukazuje obrázek 2.1,
- schéma relace – $R = R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$ [2, s. 26],
- tělo relace – $R^* \subseteq D_1 \times D_2 \times \dots \times D_n$,
- doména – jak uvádí Koch, jedná se o „pojmenovanou množinu skalárních údajů“ [2, s. 27],
- relační databáze – množina relací s určenými vazbami.



Obrázek 2.1: Relace (Zdroj: [2], s. 25)

Každé schéma relace je tvořeno alespoň jedním atributem, u kterého jsou zadány povinné vlastnosti jméno a datový typ, a dále nepovinné vlastnosti, například zda se jedná o primární nebo cizí klíč, zda má být vložena hodnota nenulová, apod.

Tělo relace pak tvoří samotná data, každý řádek pak představuje n -tici relace. Jak taková jednoduchá relace může vypadat ukazuje tabulka 2.1.

Tabulka 2.1: Ukázka tabulky relační databáze

Název atributu 1	Název atributu 2	Název atributu 3
Hodnota 1. n -tice atributu 1		Další hodnota
Další n -tice	Hodnota 2	

Integritní omezení

S relačním modelem úzce souvisí pojem **integrita**. Koch integritu definuje jako „stav, při kterém data uložená v modelu odpovídají vlastnostem objektů reálného světa“ [2, s. 28].

Existují tři základní integrity:

- doménová integrita – data daného atributu musí nabývat pouze přípustných hodnot,
- entitní integrita – každá relace musí být jednoznačně určena primárním klíčem, který musí splňovat podmínky minimálnosti a jednoznačnosti [1, s. 70],
- referenční integrita – každý neprázdný cizí klíč v relaci musí odpovídat hodnotě primárního klíče v jiné relaci [1, s. 71].

Samotná integritní omezení jsou pak „pravidla, která definují nebo omezují některé vlastnosti dat užívaných organizací“ [1, s. 71].

2.3.1 Entitně-relační modelování

Základem pro vytvoření funkční databáze je vytvořit model správně reflektující reálnou reprezentaci objektů světa do jednotlivých entit a jejich následné zakomponování do databázového schématu. Stejně tak je potřeba dodržet všechny požadavky zadání, zejména vztahy mezi jednotlivými entitami, tak jako existují v reálném životě [1, s. 155].

Entitně-relační modelování umožňuje tyto entity a jejich vztahy zobrazit pomocí diagramů bez využití technického jazyka. Existuje několik konvencí, jakým způsobem diagramy realizovat, všechny ale využívají základních geometrických objektů jako jsou obdélníky, trojúhelníky, kosočtverce a spojovací úsečky, případně šipky. Jejich základními součástmi jsou [11]:

- entita – reprezentace jednotlivého reálného objektu,

- relace – množina vztahů mezi jednotlivými entitami,
- supertypy a subtypy – vyjádření závislosti podobných entit.

2.3.2 Normalizace

Normalizace je proces, který při kterém se analyzují souvislosti mezi atributy a jejich následné shromáždění do entit a relací. Tento proces se využívá nejen při návrhu, ale při analýze chybného návrhu, a to pomocí testování dodržení tzv. **normálních** forem, které zkoumají vztahy mezi jednotlivými sloupci v tabulkách [1, s. 187].

1. normální forma

1. normální forma je pro relaci splněna právě tehdy, když neobsahuje žádné vícestupňové atributy, tedy každý atribut obsahuje právě jednu dále nedělitelnou informaci. Pokud tuto základní vlastnost nemá, je nutné provést dekompozici a rozložit informace z tohoto atributu na řádky v jiné tabulce [6, s. 18].

2. normální forma

Druhá normální forma pro relaci je splněna, pokud veškeré atributy, které netvoří primární klíč, jsou na primárním klíči funkčně závislé, a zároveň relace musí splňovat první normální formu. Dodržením této normální formy je zajištěno odstranění všech atributů, které jednoznačně nesouvisí s identifikujícím prvkem. Jednoduchým způsobem, jak vyřešit porušení druhé normální formy, je tedy přesun atributu, které výše uvedené podmínky nesplňují, do relace, kde budou záviset pouze na primárním klíči. [6, s. 20]. Tento postup se označuje jako **dekompozice**, při které musejí být splněny 3 základní podmínky, a to konkrétně musí dojít k:

- zachování všech závislostí,
- odstranění zbytečných duplicit a jiných redundantních údajů,
- relaci musí být možné zpětně spojit (např. při využití SQL příkazu JOIN).

3. normální forma

Poslední standardní normální forma řeší problematiku tzv. tranzitivní závislosti. K té dochází, pokud existuje atribut, který závisí na jiném atributu, který není primárním klíčem relace. Pro splnění třetí normální formy je proto nutné tento závislý atribut přesunout do jiné relace, kde již bude závislý pouze na primárním klíči, a ten bude v původní relaci nadále záviset výhradně na primárním klíči zkoumané relace. Kromě toho musí relace být i v druhé normální formě, a tedy zároveň i v první normální formě [6, s. 21].

2.3.3 Další normální formy

Kromě prvních třech normálních forem existují ještě následující tři, které již nejsou pro analýzu databázového systému DC podstatné, proto následuje pouze jejich výčet:

- Boyce-Coddova normální forma,
- 4. normální forma,
- 5. normální forma.

3 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

Jak již bylo uvedeno v úvodu práce, v Libereckém kraji došlo k rozhodnutí o vytvoření datového centra pro podporu plánování rozvoje sociálních služeb. Formální požadavky pro dodavatele a zpracovatele byly určeny dokumentem „Vstupní analýza podmínek pro vytvoření datového centra sociálních služeb Libereckého kraje“ (dále jen vstupní analýza) z prosince 2008.

3.1 Požadavky na vytvoření Datového centra

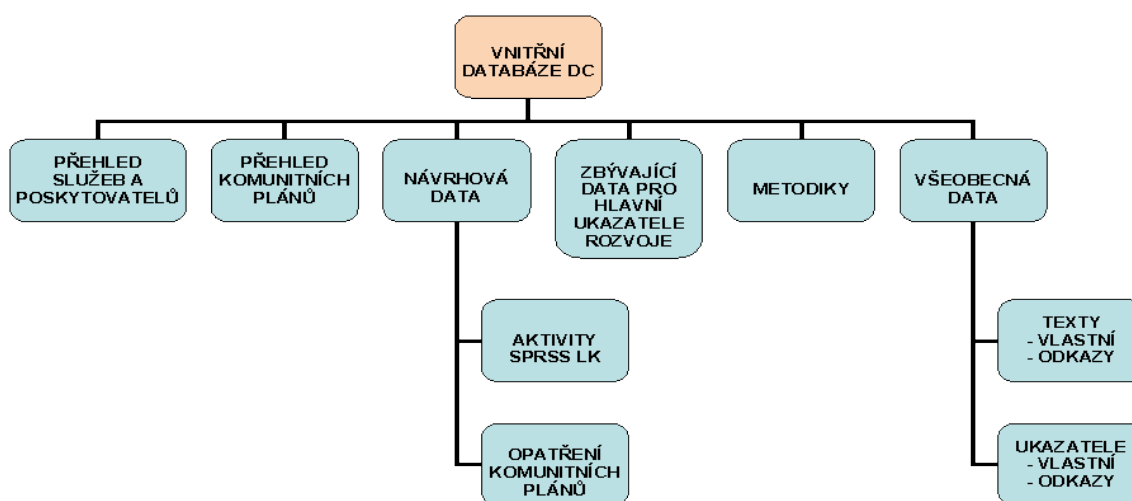
Na začátku byly definovány jasné podmínky a vstupní zdroje dat, na kterých by mělo být Datové centrum postaveno. Mezi tyto zdroje patří pět kategorií dat [5, s. 6]:

- centrálně sledovaná data – jedná se o data poskytovaná centrálními orgány, zejména MPSV. Patří mezi ně především registry sociálních služeb, které by měly sloužit jako základní stavební kámen, na kterém bude DC stavět, ale i data z dalších oblastí (regionální rozvoj, školství, ...).
- data z Libereckého kraje – tato skupina dat obsahuje informace o sociálních zařízeních v kraji, doplněné o další všeobecná data demografického charakteru.
- data poskytnutá zřizovateli sociálních služeb – jedná se zejména o obecná evidenční data, zaměřená na rozvojové plány jednotlivých sociálních služeb.
- data od poskytovatelů sociálních služeb – převážné množství dat v této skupině je tvořeno ekonomickými a evidenčními údaji.
- informace od uživatelů sociálních služeb – poslední kategorie dat je tvořena zpětnou vazbou od cílových skupin, od předchozích se odlišuje především tím, že tato data nebyla nikdy systematicky sledována a evidována.

Návrh Datového centra tedy musel počítat se sběrem dat v nejrozumnějších formátech z mnoha různých zdrojů.

Hlavními výstupy DC podle vstupní analýzy by měly být 3 okruhy [5]:

- **vytvoření a provoz DC** – jedná se o návrh struktury dat, způsobu jejich získávání z existujících zdrojů, úlohy zainteresovaných aktérů, odpovědnosti za poskytovaná data. Především analýza této problematiky je předmětem této práce.
- **poskytnutí veřejně přístupných informací z DC** – jedná se především o zpřístupnění informací formou webových stránek, propagačních tiskovin, a katalogu sociálních služeb v LK.
- **export ukazatelů pro potřeby plánování rozvoje** – tyto výstupy musí splňovat kritéria pro všechny úrovně plánování, zejména pro obce a kraje, pro poskytovatele služeb, a samozřejmě pro MPSV.



Obrázek 3.1: Návrh vnitřní struktury Datového centra (Zdroj: [4, s. 2])

Na základě definování této problematiky byly jasně uvedeny i požadavky pro zpracovatele. Ten měl za úkol kromě samotné realizace zajistit i následný provoz a údržbu, jako po stránce softwarové, tak hardwarové.

Hlavním požadavkem pro zpracovatele byly i zásady návrhu datové struktury pro Střednědobý plán rozvoje sociálních služeb (SPRSS). Prvním bylo rozdělení do dvou hlavních datových skupin:

- **hlavní ukazatele rozvoje** – tyto by měly popisovat parametry sociálních služeb a znázorňovat jejich rozvoj,

- **doplňková a všeobecná data** – tato data by měla být využitelná při analýze stavu sociálních služeb, a při sledování jejich vývoje.

3.2 Realizace a podoba Datového centra

Po ustanovení všech požadavků proběhlo výběrové řízení na firmu, která zajistí realizaci a provoz Datového centra. Toto výběrové řízení vyhrála firma Proprio, která následně v letech 2010 a 2011 navrhla základní databázovou strukturu, tak aby byly zachyceny všechny výše uvedené požadované informace. Jako softwarový základ byl použit Microsoft SQL Server, který umožňuje napojení na existující data, a zároveň poskytuje dostatečný technologický základ pro navrhovaný databázový systém. Navržená databázová struktura se dá rozdělit dle typu dat na dvě skupiny:

- **statická data** – jedná se o relační tabulky sloužící jako číselníky, a obsahující stálá a zřídka kdy modifikovaná data jako jsou např. seznam obcí, rozdělení cílových skupin, apod. Kompletní seznam těchto tabulek je uveden na straně 22 v tabulce 3.1. Tyto tabulky jsou označeny prefixem „c_“.
- **dynamická data** – obsahující konkrétní data pro jednotlivé sociální služby a zařízení, jako jejich seznam, kapacity, typ zařízení; dále pak systémové údaje jako seznam uživatelů majících přístup do systému, a definice jejich přístupových práv. Všechny tabulky spadající do této kategorie popisuje tabulka 3.2 na straně 23.

3.3 Analýza použitého řešení

3.3.1 Značení tabulek

Použitý systém značení tabulek vychází z jednoduchého principu. Typ tabulky je jednoznačně zadán jednopísmenným prefixem, odděleného znakem podtržítka. Ten může nabývat z těchto hodnot:

- c_ – číselník. Takto označené tabulky jsou převážně statického charakteru, a data, která obsahují se prakticky nemění.

Tabulka 3.1: Seznam použitých číselníků. (Zdroj: vlastní)

c_cilova_skupina	číselník cílových skupin, na které je sociální služba zaměřena
c_cilova_skupina_kategorie	číselník kategorií cílových skupin, na které je sociální služba zaměřena
c_obec	číselník obcí
c_obec_rp	číselník obcí s rozšířenou působností
c_prani_forma	číselník právních forem
c_sluzba	číselník sociálních služeb
c_sluzba_druh	číselník druhů sociálních služeb
c_sluzba_forma	číselník forem sociálních služeb
c_user_role	číselník uživatelských rolí (admin-správce, kraj, poskytovatel, obec)

d_ – datové tabulky. Tyto obsahují samotná data a ukazatele jednotlivých sociálních služeb a zařízení.

v_ – vazební tabulky, vzniklé dekompozicí relačního modelu tak, aby byla splněná třetí normální forma.

w_ – pohledy (views) určené pro zjednodušené exportování dat.

Další část názvu relační tabulky tvoří jednoznačné označení, čeho se data, která obsahuje, týkají. Tento název je většinou zapsaný česky, až na pár výjimek (např. d_user_login). Žádná tabulka neobsahuje zjevně z kompatibilních důvodů v názvu diakritiku, a mezera ve víceslovném označení je opět nahrazena znakem podtržítka.

3.3.2 Struktura číselníků

Všechny tabulky, které slouží jako číselníky, mají velice podobnou syntaxi, kterou ukazuje tabulka 3.3 na straně 24. Tato struktura je bez výjimky stejná u všech číselníků, pouze u některých je rozšířena o jeden, nebo dva atributy, jako např. kod typu *varchar(3)* u číselníku c_pravni_forma.

Tabulka 3.2: Seznam použitých relačních tabulek. (Zdroj: vlastní)

d_aktivita	přehled rozvojových aktivit SPRSS LK
d_aktivita_financovani	zdroje financování rozvojových aktivit SPRSS LK
d_aktivita_sluzba	vazební tabulka aktivity vztažené ke službám
d_metodika	přehled metodik pro plánování rozvoje sociálních služeb
d_opatreni	přehled opatření z komunitních plánů
d_plan	přehled komunitních plánů
d_plan_obec	vazební tabulka zapojených obcí do komunitního plánu
d_plan_sluzba	vazební tabulka komunitního plánu vztažené ke službám
d_poskytovatel	datová tabulka evidence poskytovatelů sociálních služeb
d_sluzba	datová tabulka sociálních služeb
d_sluzba_cilova_skupina	vazební tabulka cílových skupin na které je služba směřována
d_sluzba_kapacita	přehled kapacit sociální služby
d_sluzba_katalog	datová tabulka katalogových informací sociálních služeb
d_sluzba_kryti	přehled krytí nákladů sociální služby
d_sluzba_naklady	přehled nákladů sociální služby
d_sluzba_obec	vazební tabulka obcí, ve kterých je služba provozována
d_sluzba_pocet	hodnoty (počty) ukazatelů sociální služby
d_user	datová tabulka - evidence uživatelů, kteří mají přístup do aplikace
d_user_login	datová tabulka evidující přihlašovací procesy uživatelů
d_user_poskytovatel	definice vazby mezi uživatelem a poskytovatelem
d_user_role	definice uživatelských rolí
d_vseobecna_data	textové informace a statistické ukazatele
v_sluzba_forma	vazební tabulka pro přiřazení formy služby ke službě. Slouží jako předpis povolených vazeb
v_sluzba_rok	vazební tabulka popisující službu v jednotlivých letech, identifikátory pro jednotlivé roky

Tabulka 3.3: Struktura obecného číselníku Datového centra (Zdroj: vlastní)

klíč	Název	Typ	Délka	NULL	Poznámky
PK	Id	int		NOT NULL	Identifikátor záznamu
	Nazev	varchar	150	NOT NULL	Obecný název vlastnosti (jméno obce,...)
	PlatnostOd	datetime		NULL	Platnost záznamu od data
	PlatnostDo	datetime		NULL	Platnost záznamu do data
	Created	datetime		NOT NULL	Datum vytvoření záznamu
	CreatedBy	int		NOT NULL	Identifikátor osoby, která záznam vložila
	Updated	datetime		NOT NULL	Datum poslední úpravy záznamu
	UpdatedBy	int		NOT NULL	Identifikátor osoby, která záznam naposledy upravila
	Deleted	datetime		NULL	Datum zneplatnění záznamu
	DeletedBy	int		NULL	Identifikátor osoby, která záznam zneplatnila

Jak je vidět, všechny číselníky počítají s možnými úpravami, a též jsou přizpůsobeny na jednoduchou informaci, kdo a kdy naposledy záznam upravil, což je ale u číselníku tvořeného převážně statickým obsahem velice diskutabilní vlastnost. Obdobně je též otázkou, zda má význam uvažovat platnost názvu obce v číselníku `c_obec`, resp. `c_obec_rp`.

3.3.3 Struktura datových tabulek

Datové tabulky Datového centra jsou již více komplexní než číselníky, přesto základní struktura je stejná. Každý z nich tedy obsahuje informace o platnosti, vytvoření jednotlivých záznamů, jejich úpravě, případně o jejich zneplatnění.

Podívejme se nyní na hlavní datovou tabulku, `d_sluzba`. Kromě standardní sady atributů obsahuje následující:

Tabulka 3.4: Další atributy tabulky d_sluzba (Zdroj: vlastní)

Název atributu	Typ	Původní poznámky	Komentář
Cislo	varchar(7)	Číslo služby	
RegistraceDatum	datetime	Datum registrace	
PoskytovatelId	int	Identifikátor poskytovatele	
SluzbaFormaId	int	Identifikátor služby a její formy	
PocetId	int	Identifikátor hodnot ukazatelů služby	Lze spočítat pomocí funkce COUNT
NakladyId	int	Identifikátor nákladů služby	
Kapacita Id	int	Identifikátor kapacity služby	
KrytiId	int	Identifikátor krytí nákladů na službu	
InspekceDatum	varchar(10)	Datum provedené inspekce	Závažná chyba, navrhovatel mylně předpokládá, že inspekce může být pouze jedna, a navíc zvolil nevhodný datový typ, zde by byl optimální datetime
InspekceVysledek			Analogicky jako u InspekceDatum
RozvojPlan	varchar(1000)	Rozvojový plán služby (jeho aktuální popis)	
Provoz	varchar(300)	Provozní doba	
Cena	varchar(200)	Cena služby nebo odkaz na informace o její ceně	
UkazatelTyp	char(1)	Typ sledovaných ukazatelů	Zkrácení typu ukazatele na jedno písmeno je zajímavý přístup, ale v takto komplexním systému by bylo vhodnějším použít vazbu na číselník

4 VLASTNÍ NÁVRHY ŘEŠENÍ

V přecházející části, zabývající se analýzou stavu Datového centra k prosinci roku 2011 byly uvedeny základní nedostatky databázového návrhu. V této kapitole bych rád tyto nedostatky podrobně rozebral a navrhl řešení, jak je eliminovat. Mezi tyto chyby návrhu patří zejména:

- míchání českých a anglických označení,
- nedokonalý návrh atributů, zejména jejich výchozích hodnot a nastavení vlastnosti NOT NULL,
- špatné rozlišení typů tabulek.

4.1 Normalizace

V této části navrhnu změny, které zajistí normalizaci celého systému, a to podle běžných tří normálních forem.

4.1.1 Dodržení první normální formy

První normální forma zajišťuje, aby pro každý záznam existovala pro každý atribut maximálně jedna přiřazená hodnota. V potaz uvažuji pouze fyzické tabulky, a nikoliv pohledy (views), ty už ze své podstaty agregace dat musí umožnit zobrazit více údajů stejného typu v jedné větě.

Protože všechny tabulky číselníků mají téměř identickou strukturu, je velice jednoduché odhalit, že všechny první normální formu splňují. Nikde se nenachází žádný atribut, který by umožnil přiřadit více hodnot pro jeden řádek entity.

U datových tabulek je základní struktura také podobná, stejně jako u číselníků. Z těchto atributů všechny základní první normální formu také splňují, jelikož atributy Id, Created, CreatedBy, PlatnostOd, PlatnostDo, Updated, UpdatedBy, Deleted a DeletedBy nabývají hodnot typů int a datetime, proto zde nevzniká prostor pro uložení více hodnot.

Nyní se podívejme podrobněji na jednotlivé tabulky, kde se již problémy s porušením první normální formy nacházejí. První tabulkou takovou tabulkou je `d_plan`, která nese informace o přehledu komunitních plánů. Zde věnujme pozornost zejména dvěma atributům: `PocetSkupin` (typ `smallint` a `NazvySkupin` (typ `varchar(1 000)`). Toto řešení je zcela nevhodné a v rozporu s první normální formou. Informace o pracovních skupinách by proto měla být umístěna do samostatné tabulky:

```
CREATE TABLE d_pracovni_skupiny (  
    Id int NOT NULL,  
    PlanId int NOT NULL,  
    NazevSkupiny varchar(200) NOT NULL,  
    Created datetime,  
    CreatedBy int,  
    Updated datetime,  
    UpdatedBy int,  
    Deleted datetime,  
    DeletedBy int,  
    PRIMARY KEY (Id)  
)
```

Počet pracovních skupin lze pak místo původního atributu `PocetSkupin` zjistit a implementovat jednoduchým SQL dotazem s využitím funkce `count`:

```
SELECT count(Id) FROM d_pracovni_skupiny  
WHERE d_pracovni_skupiny.PlanId = <číslo>
```

Další tabulkou, kde se nachází částečné porušení první normální formy, je tabulka `d_poskytovatel` a atribut `sídlo`. Toto je častý problém, protože obsahuje více hodnot, a to ulici, číslo popisné, město a poštovní směrovací číslo. Ale vzhledem k tomu, že tento údaj není pro zlepšení úrovně poskytování sociálních služeb kriticky důležitý, a na výkon celého databázového systému nebude mít příliš velký vliv, lze zde vzhledem k rozsahu databáze ponechat typ `varchar (150)`.

Největší a nejzásadnější tabulkou je kompletní seznam služeb označený jako `d_sluzba`. Přes komplexnost a rozsáhlost tabulky se zde opět nachází porušení první normální formy, a to konkrétně v následujících atributech:

- **Provoz** – tento atribut je typu `varchar(300)` a může obsahovat téměř libovolný popis otevírací doby. Pokud má být ale užitečný pro automatizované exportování (např. na WWW), měla by být provozní doba uvedena v samostatné tabulce.
- **Cena** – dle popisu obsahuje ceník, a to buď přímo s uvedenými hodnotami, nebo formou odkazu na externí dokument. Tento způsob by vzhledem k různorodosti dat mohl být zachován, přesto by bylo dobré při importu dat zajistit jejich integritu, a údaje o cenách shromáždit ve zvláštní tabulce, podobně jako u atributu `Provoz`.
- **Adresa** – učebnicový příklad porušení první normální formy. Aby se daly jednotlivé subjekty roztrždit například podle měst, navrhuji použít speciální tabulku `d_adresy` se strukturou uvedenou v tabulce 4.1 na straně 28.
- **Působnost** – působnost služby je vždy definována podle krajů, bylo by proto vhodné nahradit slovní popis opět samostatnou tabulkou vazeb napojenou na číselník se seznamem jednotlivých krajů.

Všechny údaje uvedené v předchozím výčtu platí i pro tabulku `d_sluzba_katalog`, je-li má prakticky totožnou strukturu.

Tabulka 4.1: Struktura navrhované tabulky `d_adresa` (Zdroj: vlastní)

Název atributu	Typ atributu	Další vlastnosti atributu
IdAdresy	int	NOT NULL, primární klíč
IdSluzby	int	NOT NULL, cizí klíč
JePrimarni	bit	NOT NULL, Default = 1
Ulice	varchar (50)	
CisloPopisne	varchar (7)	
Mesto	varchar (30)	
PSC	varchar (5)	

Poslední tabulkou, která odporuje první normální formě je `d_vseobecna_data`, obsahující statistické ukazatele a další převážně textové informace o jednotlivých obcích. Všechny popisné atributy (`Geo`, `Cleneni`, `Socio` a `Vazby`) mají typ `varchar(max)` a obsahují velké množství údajů. Přesto, pro účely Datového centra je tento formát více než dostatečný a informace zde uvedené nebudou strojově zpracovávány, slouží především pro zpracování do nejrůznějších propagačních materiálů, metodických příruček a dalších materiálů a proto lze zde porušení první normální formy ignorovat a nechat výše čtyři jmenované atributy jako textové informace.

4.1.2 Dodržení druhé normální formy

Splnění druhé normální formy je založené převážně na podmínce, aby tabulka měla složený primární klíč. Vzhledem k tomu, že všechny tabulky mají jednoduchý primární klíč tvořený jedním atributem, lze tvrdit, že tabulky splňující první normální formu jsou i v druhé normální formě.

4.1.3 Dodržení třetí normální formy

Třetí normální forma se snaží o eliminaci tranzitivní závislosti, která ale byla brána na vědomí již při návrhu systému. Přesto se v systému objevuje několik výskytů. První z nich je součástí většího problému, kterým je atribut `KontaktniOsoba`. Zde je velká šance, že více subjektů poskytujících sociální služby bude mít stejnou kontaktní osobu. Proto by kontaktní osoby a veškeré atributy odkazující se na osoby měly by tvořeny jako typ `int` a odkazovat se do samostatné tabulky osob, kde se může v případě potřeby jednoduše upravit libovolný údaj.

Druhým výskytem jsou tabulky obsahující atributy `Mesto` a `PSC`. Vzhledem k tomu, že město je přímo závislé na PSČ, a naopak, lze vytvořit číselník, kde primárním klíčem bude atribut `PSC` typu `varchar(5)`. Tím odstraníme tranzitivní závislost a všechny inkriminované tabulky budou splňovat třetí normální formu.

4.2 Označení tabulek

Názvy jednotlivých tabulek použitých v databázovém systému Datového centra byly podrobně popsány v kapitole 3.3.1 na straně 21.

Jedním z problémů je jazyková nejednotnost v názvech, kde dochází k míchání českých a anglických názvů, a v jednom případě (`d_user_poskytovatel`) dokonce k jejich kombinování.

Další nesrovnalostí je uložení systémových údajů v tabulkách, které jsou označeny prefixem `d_`, tedy datové. Bylo by proto vhodné tyto tabulky odlišit novým prefixem `s_` (systémové).

Užití prefixů pro typ tabulky je vhodné řešení, ale nereflektuje plně potřeby systému a jeho současné využití. Aby byla dodržena logicky navržená forma názvů, navrhuji přejmenovat některé tabulky způsobem uvedeným v tabulce 4.2.

4.3 Přejmenování atributů

Označení jednotlivých atributů napříč databází je konzistentní, a je založené na základních pravidlech SQL jazyka pro jejich názvy. Z toho vyplývá několik použití pouze velkých a malých písmen, kde velké písmeno vždy označuje nové slovo, a eliminuje tak nemožnost použití znaku mezery (, „). Bylo by možné též místo toho použít místo znaku podtržítka (, _), ale zvolené řešení je účelné a funkční. Osobně bych zvolil všechny názvy uvedené malými písmeny s použitím podtržítka. Jednotlivé názvy pak více vyniknou ve zdrojovém kódu jazyka SQL a mohou usnadnit budoucí úpravy.

Přesto jsem v označení atributů objevil nesourodost, kterou je míchání českého a anglického jazyka. Veškeré informační atributy (např. `Sluzba`, `Provoz`, a další) jsou označeny česky, kdežto všechny atributy nesoucí informace o modifikaci dat (`Created`, `Updated`, `Deleted`, `CreatedBy`, ...) jsou označeny anglicky. Jsou tak tedy částečně rozlišeny od informačních atributů, ale přinášejí tak zmatek a snižují přehlednost při budoucích úpravách, které mohou být prováděny i jinými stranami, než návrhovatelem struktury databáze. Z tohoto důvodu navrhuji upravit tyto atributy tak, aby byly vedeny ve stejném jazyce, tj. česky, a jednoznačně odlišit, že se jedná o informace, které nene-

sou data týkající se přímo sociálních služeb v Libereckém kraji. Toto odlišení lze provést např. přidáním prefixu *S_* (systémové informace).

Poslední nesrovnalostí týkající se výše vyjmenovaných atributů je označení *Deleted (By)*. Tento atribut by podle názvu měl označovat, kdy byl přiřazený záznam smazán, a kým byl smazán, což je už z principu nesmyslné, u smazaného záznamu logicky nelze udržet informaci o jeho odstranění. Ovšem po nahlédnutí do původní dokumentace a popisu objevíme, že tento atribut označuje, kdy byl záznam zneplatněn, a kdo jej jako neaktuální označil. Proto by bylo vhodné i tuto informaci v názvu atributu označit korektně.

Kompletní návrh změn názvů atributů přehledně ukazuje tabulka 4.3.

4.4 Řízení přístupů k DB

Jednou z dalších sporných částí celého databázového systému je systém řízení přístupu k datům a audit jednotlivých změn. Ten je v době analýzy systému zajištěn pomocí šesti atributů ve všech tabulkách. Jejich seznam s navrhovanými novými lokalizovanými názvy ukazuje tabulka 4.3, popis vlastností atributů je uveden v analytické části práce v tabulce 3.3 na straně 24. Z těchto tabulek je vidět, že pro každý záznam se uchovávají následující informace a data:

- datum vložení záznamu do databáze spolu
- datum poslední změny
- datum smazání záznamu

Ke každému z těchto atributů je zároveň přiřazen další atribut typu *int* stejného jména se suffixem *By*, který definuje ID uživatele odpovědného za provedenou změnu.

V případě vložení nového záznamu je tento způsob dostatečný, jelikož máme v systému kontrolu nad tím, kdy a kým byl záznam přidán, nelze jeden záznam s unikátním identifikátorem vložit vícekrát.

Problém nastává u druhého atributu, *Updated (By)*. Ten nám umožní uložit pouze poslední provedenou úpravu právě jedním uživatelem. To je ovšem velice nepraktické, nedá se tak zpětně vystopovat žádná další změna. I když účel celého databázového

systemu nepředpokládá častou úpravu dat, je potřeba s ní počítat, zejména z důvodů pravidelné aktualizace ekonomických a statistických dat na základě výměny informací mezi poskytovateli a dalšími zainteresovanými subjekty (MPSV, FÚ, obecní úřady, apod.)

U posledního atributu označujícího zneplatnění dat, je situace analogická jako u vkládání záznamu, záznam se stane neplatným většinou právě jednou. Teoreticky zde ale může nastat situace, že např. některému poskytovateli byla dočasně pozastavena licence na provozování sociálních služeb, ten v krátké době důvod tohoto pozastavení eliminuje, licence mu bude při nejbližší aktualizaci dat již navracena a záznam v databázi by mohl být opět označen jako platný. Tuto možnost lze realizovat i při současném návrhu, při vložení záznamu se hodnota atributu `s_zneplatneno` nastaví na hodnotu `NULL` a při zneplatnění dat se nastaví aktuální datum, ale nelze zde samozřejmě sledovat historii takových úprav.

Hromadným řešením výše uvedených problémů je modifikace stávající struktury databáze takovým způsobem, který umožní sledovat historii jednotlivých změn pro každý záznam v tabulce a přiřadit k němu odpovídajícího pracovníka. Aby byly splněny jednotlivé normální formy, musela by existovat pro každou existující datovou tabulku další auditní tabulka, shromažďující zejména následující údaje:

- ID odpovědného uživatele (případně hodnota nula, pokud se bude jednat o systémovou úpravu)
- typ prováděné změny – odkaz na číselník
- čas změny
- komentář – možnost pro doplnění informací a shrnutí změn

Vzhledem k množství údajů, které by bylo nutno tímto způsobem uchovávat, bych přehodnotil, pro které tabulky je potřeba seznam změn udržovat. V principu se jedná o tabulky, které budou upravovat zaměstnanci zainteresovaných subjektů, ať už pracovníci z MPSV, či jednotlivých subjektů z řad poskytovatelů zasílajících pravidelně aktualizované údaje ekonomického, či statistického charakteru. Určitě nás tedy nebude zajímat změna v jednotlivých tabulkách číselníků, a obdobná situace nastane i v případě vazebních tabulek.

Do seznamu tabulek, u kterých by bylo vhodné sledovat změny, a pro které by bylo potřeba vytvořit auditní tabulky, bych zařadil následující:

- d_aktivita
- d_aktivita_financovani
- d_metodika
- d_opatreni
- d_plan
- d_poskytovatel
- d_sluzba
- d_kapacita
- d_sluzba_kryti
- d_sluzba_naklady
- d_sluzba_pocet
- d_vseobecna_data

Obecnou strukturu auditní tabulky uchovávající údaje o změnách, by bylo možné vytvořit pomocí SQL příkazu uvedeného pod tímto odstavcem. Tyto auditní tabulky by rovněž měly být odlišeny prefixem, zde budu užívat a..

```
CREATE TABLE a_sluzba (  
    id int IDENTITY(1),  
    id_uzivatele int NOT NULL,  
    id_sluzby int NOT NULL,  
    typ_zmeny smallint NOT NULL,  
    kdy datetime NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY typ_zmeny REFERENCES c_typ_zmeny (typ_zmeny),
```

```

FOREIGN KEY id_sluzby REFERENCES d_sluzba (Id)
FOREIGN KEY id_uzivatele REFERENCES d_user (Id)
)

```

A samozřejmě ještě před tím nesmíme zapomenout vytvořit číselník, obsahující jednotlivé typy změn, aby byla splněna vazba cizího klíče:

```

CREATE TABLE c_typ_zmeny (
    typ_zmeny int IDENTITY(1),
    popis_zmeny varchar(50)
)

INSERT INTO c_typ_zmeny (popis_zmeny) VALUES
    ('vložení'),
    ('úprava'),
    ('zneplatnění'),
    ('obnovení platnosti záznamu')

```

Aby výše uvedené navrhované změny byly jednoduše použitelné v praxi v existujícím systému, je potřeba zajistit, aby se při libovolné změně dat zapsal příslušný řádek do auditní tabulky. Tuto funkcionalitu můžeme zajistit dvěma základními způsoby:

- s použitím triggeru, který je zavolán při změně dat nad sledovanou tabulkou,
- implementací přímo v aplikaci pro koncové uživatele.

Pro ilustraci přikládám ukázkový trigger pro tabulku `d_sluzba` který do auditní tabulky `a_sluzba` zapíše informace o provedené změně:

```

CREAE TRIGGER t_sluzba_novy ON d_sluzba
    FOR INSERT, UPDATE
AS
    DECLARE @typ AS smallint;
    DECLARE @id_upravene_sluzby as int;
    SET @typ = 1; -- nastavení výchozího typu změny na 'vložení'
    SET @id_upravene_sluzby = SELECT Id FROM inserted

```

```
IF EXISTS (SELECT * FROM deleted)
BEGIN
    SET @typ =
        CASE
            WHEN EXISTS (SELECT * FROM inserted) THEN 2
            ELSE 3
        END
END
INSERT INTO a_sluzba (id_uzivatele, id_sluzby, typ_zmeny, kdy) VALUES
    (USER_ID(), @id_upravene_sluzby, @typ, GetDate())
GO
```

Tabulka 4.2: Nové navrhované názvy tabulek (Zdroj: vlastní)

Původní název tabulky	Navrhovaný český název tabulky s opraveným prefixem	Důvod změny názvu
d_user	s_uzivatel	počeštění + změna prefixu na systémový
d_user_poskytovatel	v_uzivatel_poskytovatel	počeštění + změna prefixu na vazební
d_user_login	s_uzivatel_prihlaseni	počeštění + změna prefixu na systémový
d_user_role	s_uzivatelske_role	počeštění
d_user_role	s_uzivatelske_role	počeštění + změnu prefixu na systémový
d_sluzba_obec	v_sluzba_obec	změna prefixu na vazební
d_sluzba_cilova_skupina	v_sluzba_cilova_skupina	změna prefixu na vazební
d_plan_sluzba	v_plan_sluzba	změna prefixu na vazební
d_plan_obec	v_plan_obec	změna prefixu na vazební
d_aktivita_sluzba	v_aktivita_sluzba	změna prefixu na vazební

Tabulka 4.3: Nové označení atributů nesoucích informace o modifikaci dat (Zdroj: vlastní)

Původní název atributu	Český název	Nový název s použitím minuskul
Created	S_VytvorenoKdy	s_vytvoreno_kdy
CreatedBy	S_VytvorenoKym	s_vytvoreno_kym
Updated	S_UpravenoKdy	s_upraveno_kdy
UpdatedBy	S_UpravenoKym	s_upraveno_kym
Deleted	S_ZneplatненоKdy	s_zneplatnenno_kdy
DeletedBy	S_ZneplatnennoKym	s_zneplatnenno_kym

ZÁVĚR

Ve své bakalářské práci jsem nejprve v úvodní části představil problematiku sociálních služeb v České republice a důvody, které vedly k vytvoření registru těchto služeb, a následně Datového centra sociálních služeb libereckého kraje.

V následující části popisující teoretické pozadí problematiky systémů řízení báze dat, historický vývoj a rozdělení používaných datových modelů. U relačních databázových modelů jsem důkladněji popsal používané principy pro optimalizaci databázových systémů, zejména normalizaci a využití integritních omezení.

Hlavní částí této práce je analýza stavu relačního modelu použitého v Datovém centru. Zde jsem nejprve uvedl požadavky, které byly kladeny na Datové centrum, a zda nakonec byly ve fázi realizace splněny. Poté jsem zkoumal především dodržení normálních forem, a dále využití jednotlivých relací a jejich atributů, a to tak, aby bylo dosaženo maximální užitečnosti pro sociální služby a pro budoucí úpravy použitého systému.

V poslední části jsem uvedl konkrétní příklady jak opravit nalezené chyby a jak vylepšit funkcionalitu a rozšiřitelnost databázového systému. Sem lze zařadit především možnost využití částečného verzování a auditních funkcí za využití standardních technologií používaných v SQL jazyce, zejména triggerů.

Celkově lze tuto práci tak shrnout, že použitý databázový systém slouží svému účelu a může dobře podporovat rozhodování o sociálních službách. Ale jakákoliv změna provozovatele bude provázena dlouhým studováním použitého řešení, které není kompletně jednotné.

Doufám, že tato práce poslouží jako inspirace pro provozovatele Datového centra, a v případě budoucích úprav dojde k realizaci alespoň některých mnou navrhovaných změn, které by měly usnadnit následný provoz a údržbu.

SEZNAM POUŽITÉ LITERATURY

- [1] CONOLLY, T., BEGG, C., HOLOWCZAK, R. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. 584 s. ISBN 978-802-5123-287.
- [2] KOCH, M., NEUWIRTH, B. *Datové a funkční modelování*. 3. vyd. Brno: Akademické nakladatelství CERM, 2008. 121 s. ISBN 978-802-1437-319.
- [3] LAUERMAN, J., BENEŠOVÁ, I. *Krajská metodická příručka plánování sociálních služeb v Libereckém kraji*. Praha: Vysoká škola regionálního rozvoje, 2010. 80 s. ISBN 978-80-87174-01-2.
- [4] LAUERMAN, J. *Návrh struktury Datového centra sociálních služeb Libereckého kraje*. Liberec: LB plán, 2010. 3 s.
- [5] LAUERMAN, J. *Vstupní analýza podmínek pro vytvoření Datového centra sociálních služeb Libereckého kraje*. Liberec: JL-plán, 2008. 25 s.
- [6] OPPEL, A. *SQL bez předchozích znalostí*. Brno: Computer Press, 2008. 240 s. ISBN 978-80-251-1707-1.
- [7] *Zákon č.108/2006 Sb., o sociálních službách, ze dne 14. března 2006*.
- [8] BATKO, M. *Relační vs. objektově-relační vs. objektové databáze 2000* [cit. 2012-03-02]. Dostupné z [www http://www.fi.muni.cz/~xbatko/oracle/compare.html](http://www.fi.muni.cz/~xbatko/oracle/compare.html)
- [9] KOCAN, M. *Databáze nejsou jen relační, díl nultý 2001* [cit. 2012-02-25]. Dostupné z [www http://www.dbsvet.cz/view.php?cisloclanku=2001111201](http://www.dbsvet.cz/view.php?cisloclanku=2001111201)
- [10] MINISTERSTVO PRÁCE A SOCIÁLNÍCH VĚCÍ. *Sociální služby*. [cit. 2011-12-14]. Dostupné z [www http://www.mpsv.cz/cs/9](http://www.mpsv.cz/cs/9)
- [11] ŠIROKÝ, J. *Logická reprezentace dat*. 2004 [cit. 2012-04-12]. Dostupné z [www http://homen.vsb.cz/~sli95/ISVDAS/IS/IS_model_dat_2.htm](http://homen.vsb.cz/~sli95/ISVDAS/IS/IS_model_dat_2.htm)

- [12] TORVALDS, L. <torvalds@linux-foundation.org>. *Re: Linux 2.6.39-rc3*. 2011-04-13 [cit. 2011-12-13]. Dostupné z [www http://thread.gmane.org/gmane.linux.kernel/1126136](http://thread.gmane.org/gmane.linux.kernel/1126136)
- [13] ŽÁK, K. *Historie relačních databází*. 2001 [cit. 2011-12-14]. Dostupné z [www http://www.root.cz/clanky/historie-relacnich-databazi/](http://www.root.cz/clanky/historie-relacnich-databazi/)

SEZNAM OBRÁZKŮ

2.1	Relace	15
3.1	Návrh vnitřní struktury Datového centra	20

SEZNAM TABULEK

2.1	Ukázka tabulky relační databáze	16
3.1	Seznam použitých číselníků	22
3.2	Seznam použitých relačních tabulek	23
3.3	Struktura obecného číselníku Datového centra	24
3.4	Další atributy tabulky d_sluzba	25
4.1	Struktura navrhované tabulky d_adresa	28
4.2	Nové navrhované názvy tabulek	36
4.3	Nové označení atributů nesoucích informace o modifikaci dat	37

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

COBOL Common Business-Oriented Language

CodasyI Conference on Data Systems Languages

DC Datové centrum sociálních služeb Libereckého kraje

DBTG Database Task Group

FÚ Finační úřad

LK Liberecký kraj

MPSV Ministerstvo práce a sociálních věcí

SPRSS Střednědobý plán rozvoje sociálních služeb

SQL Structured Query Language

SŘBD systém řízení báze dat